سید محمد طاها طباطبایی – تمرین سری سوم
۹۸۱۲۷۶۲۸۳۸

چکیده:
در این تمرین، تمرکز بر روی پیاده‌سازی فیلتر های مختلفی است که با اعمال آن ها روی تصاویر، بتوان کیفیت عکس را با روش های حذف نویز و برجسته سازی لبه ها، بهبود داد. فیلتر های میانگین و میانه، برای حذف نویز موثر هستند. ماسک های وزن دار مانند لاپلاسین و سوبل، اگر با تصویر کانوالو شوند، لبه های تصویر را جدا می‌کنند.

توضیحات فنی:

۳.۱.۱

به این دلیل که وزن همه پیکسل ها در این فیلتر، یکسان است، که باعث می‌شود، تاثیر لبه ها کمتر دیده شود، یا پیکسل های حاشیه ای به اندازه پیکسل های مرکزی تاثیرگذار باشند. همچنین در همه شرایط، میانگین گیری برای حذف نویز روشی کارآمد نیست، مثل نویز فلفل نمکی، پس برای حذف نویز نیست کاملا بهینه نیستند.

۳.۱.۲

خیر، بالعکس، تاثیر آن تثبیت می‌شود.

۳.۱.۳

با تکرار اعمال فیلتر روی تصویر، هربار تصویر مقدار بیشتری blur می‌شود. در کد این بخش، عملیات اعمال فیلتر را ۱۶ بار تکرار کرده ایم، که مشاهده می‌شود، هر بار تصویر مقداری مات تر می‌شود، و در نهایت در تکرار ۱۶ ام، تصویر نسبتا blur شده ای داریم. تابع box_filter، برای ایجاد باکس فیلتر و تابع clip_filter برای ایجاد پدینگ از نوع clip filter است. درتابع clip، ابتدا یک فرم با ابعاد مورد نیاز که درواقع، ابعاد عکس اصلی + پدینگ ها باشد، می‌سازیم. در ادامه، پیسکل های عکس اصلی را مطابق جایگاه جدیدشان در فریم، به فریم منتقل می‌کنیم. تابع دوم استفاده شده، box_filter، به این شکل تعریف می‌شود که در ورودی، تصویر ورودی، سایز پنجره فیلتر مورد نظر و ابعاد پدینگ عکس مشخص می‌شود. پس از اینکه یک آرایه جدید جهت نگهداری عکس ساخته شده در این تابع با نام newImage می‌سازیم، در دو حلقه تو در تو، به اندازه پنجره مشخص شده، روی تصویر ورودی حرکت می‌کنیم. در هر مرحله، میانگین پیکسل های موجود محاسبه و در عکس جدید گذاشته می‌شود.

۳.۱.۴

اعمال فیلتر با سایز بزرگتر، موجب می‌شود تا شدت blur شدن تصویر، بیشتر شود، به طوری که در کد این بخش، با استفاده از پنجره ای به سایز ۷، توانستیم با ۱۲ گام اعمال فیلتر، تصویر blur تری نسبت به مرحله قبل که ۱۶ بار فیلتر را اعمال کردیم، به دست آوریم. با توجه به نتایج، مشاهده می‌شود که هرچه سایز فیلتر بزرگتر شود، مقدار بلر شدن بیشتر می‌شود، اما کاهش نویز بیشتری هم خواهیم داشت.

۳.۱.۵

در این بخش، ۳ سایز مختلف پنجره را امتحان کرده ایم، سایز ۳ blur کمی را ارائه می‌دهد، در حالی که blur با اندازه ۷، بسیار شدید است. به نظر، سایز پنجره ۵، برای استفاده مناسب تر است.

۳.۱.۶

انتظار می‌رود تا پس از میانگین گیری اعمال شده و blur شدن تصویر، با چندبار اعمال فیلتر laplacian، تصویر هر بار نویزدارتر شود. تابع Laplacian_filter، برای اعمال فیلتر لاپلاسین استفاده می‌شود. در این تابع، از یک تابع بیسیک تر به نام weighted_filter استفاده شده است که به طور کلی تر، هر نوع ماسکی را روی تصویر ورودی اعمال می‌کند.

۳.۲.۱

برای اجرای این بخش، تابع salt_pepper برای تولید نویز فلفل نمکی و تابع median_filter برای پیاده‌سازی فیلتر میانه نوشته شده است. در تابع salt_pepper، ابتدا برحسب درصد نویز مورد نیاز، تعداد پیکسل هایی که باید تبدیل به نویز شود را حساب می‌کنیم. سپس، به طور رندوم، یک نویز فلفل یا نک تولید می‌شود. در ادامه، یک پیسکل رندوم انتخاب می‌شود، و اگر این پیکسل، قبلا تبدیل به نویز نشده باشد، تبدیل به نویز نمک یا فلفل می‌شود.

حال در ابتدا، با شدت های مختلف، نویز فلفل نمکی تولید و تصاویر ساخته شده را نمایش می‌دهیم.

در سلول بعدی، برای یک تصویر با نویز فلفل نمکی با شدت ۰.۲، فیلتر میانه را اعمال می‌کنیم تا نویز را حذف کند.

در ادامه، طبق خواسته صورت سوال، برای شدت نویز های متفاوت و پنجره های متفاوت، تاثیر حذف نویز را بررسی میکنیم. مشخص می‌شود سایز پنجره کوچک‌تر، چون به شکل محلی تری پردازش می‌کند، برای حذف نویز کارآمد تر است.

	3	*5*	*7*	*9*
0.05	3738.7139	7040.7324	10308.9375	13479.5914
0.1	16138.5199	21090.2548	25748.6226	30040.8696
0.2	35226.9161	40978.3683	45420.1432	48982.7334
0.4	52018.7363	54163.8969	55536.1168	56176.4510

عملکرد تابع median_filter، استفاده شده در این بخش، مشابه box_filter است که بالاتر توضیح داده شد، با این تفاوت که عملیات انتخاب میانه در این تابع، جایگزین میانگین گیری می‌شود.

۳.۲.۲

در سلول اول، نویز گوسین را به تصویر اضافه می‌کنیم. برای اینکار پارامتر های لازم را مشخص می‌کنیم، و از تابع رندوم برای ساخت نویز گاوسی کمک می‌گیریم.

در سلول بعدی، تصویر را با فیلتر میانه، نویزگیری می‌کنیم.

در سلول سوم، تصویر را با فیلتر میانگین نویزگیری می‌کنیم.

با توجه به اعداد به دست آمده در محسابه MSE، مشاهده می‌شود که به طور کلی عملکرد فیلتر میانه بهتر است.

۳.۲.۳

در این بخش، ابتدا نویز گاوسین وسپس نویز فلفل نمکی به عکس افزوده شده است، سپس به ترتیب، فیلتر میانه و میانگین روی تصویر اعمال شده است. دلیل اینکار این است که اعمال فیلتر میانگین روی نویز گاوسی، اثر کمتری دارد، پس در مرحله دوم اعمال می‌شود، تا ابتدا با فیلتر میانه، اثر نویز گاوسی که شدید تر از فلفل نمکی است را کاهش دهیم.

۳.۳.۱

عکس ورودی با کمک روش ماسکینگ غیرشارپ بهبود یافته است. برای عکس مورد نظر، با آزمون و خطا، سایز پنجره را ۱۱ و ضریب را ۰.۰۲۰ درنظر گرفتیم. مشاهده می‌شود که عکس بهبود می‌یابد. به طور مثال، شاخه های درخت سمت چپ تصویر قابل تمایزتر می‌شود، و خطوط کاشی ها در قسمت سمت راست تصویر که تاریک تر است، واضح تر می‌شود.

۳.۴.۱

در این بخش، از تابع weighted_filter برای پیاده‌سازی فیلتر های وزن دار جهت پیدا کردن و تقویت لبه ها استفاده شده است. تصاویر خروجی، لبه ها و تصاویر حاصل از افزودن لبه ها به تصاویر اصلی هستند. در بحث پیدا کردن لبه های عمودی، مشخصا سوبل بهتر از دو فیلتر دیگر عمل کرده است، هرچند تصویر حاصل از افزودن لبه های سوبل به تصویر اصلی، چون لبه ها شدیدتر هستند، حالت نویزی و کمی خراب پیدا کرده است. دلیل بهتر بود سوبل نسبت به دو فیلتر دیگر نیز، یک این است که در سه سطر اختلاف را محاسبه می‌کند(برخلاف حالت اول) و دوم اینکه شدت تغییرات پیکسل ها در سطر دوم که به مرکز نزدیک تر است، بیشتر است(برخلاف حالت دوم).

۳.۴.۲

در این بخش، از همان تابع قبل، اینبار با ماسک هایی، مطابق با ضرایب روبرت استفاده کرده ایم. ماسک اول، روبرت در جهت نیم‌ساز ناحیه دوم و چهارم محور مختصاتی، و ماسک دوم، روبرت در جهت نیم‌ساز ناحیه اول و سوم است.

در مورد مقایسه کیفیت این فیلتر نسبت به فیلتر های مرحله قبل، با توجه به اینکه میتوانیم با دو بار اعمال فیلتر در جهت های گفته شده، هم لبه های قطری را داشته باشیم، و هم اگر نتیجه این دو را با هم جمع کنیم، لبه های افقی و عمودی نیز با تقریب مناسبی حاصل می‌شوند، احتمالا در استفاده عملی، بهتر از پیدا کردن لبه ها در جهت های عمودی و افقی باشند، اما از لحاظ کیفیت، ضرایب سوبل در جهت عمودی لبه های شارپ تری را مشخص کرد.

۳.۵.۱

در این بخش مطابق فرمول، عملیات ماسکینگ تقویت بالا را انجام می‌دهیم. عملیات، مطابق فرمول صورت سوال انجام شده است و توضیح خاصی ندارد. با توجه به نتایج خروجی به‌دست آمده، به نظر می‌رسد، اندازه پنجره ۹ یا ۱۱ و ضریب ۰.۰۲ بهترین انتخاب باشد، زیرا برای اندازه پنجره های پایین تر، شارپینگ ضعیف تر است و برای ضریب شدت های بالاتر، نویز به تصویر اضافه می‌شود. ضرایب بسیار نزدیک به صفر نیز تاثیر شارپ کنندگی زیادی ندارند.